

## **Portable and Professional Audio Converters**

### **AIC Script Language**

A script is simply a text file that contains data to send to the serial control buses. The scripting language is quite simple. Each line in a script file is one command. There is no provision for extending lines beyond one line. A line is terminated by a line feed (0x0a) or a carriage return and line feed (0x0d0a). Characters can be either upper or lower case as all are converted to lower case (except for the string text in the Break command).

Script language files can be created by users with text editors and by the PurePathStudio assembler. The files are used as input by the codec EVM GUI's and the device drivers of supported operating systems. The codec EVM GUI's send the commands through a USB device to the codec's command port. The device drivers will send the command directly to the codec's command port.

The first character of a line is the command. Commands are:

- # Comment
- r Read from the serial control bus
- w Write to the serial control bus
- > Extend repeated write commands to lines below a w
- i Set interface bus to use (GUI only)
- b Break (GUI only)
- d Delay
- f Wait for Flag

### **Command Syntax**

#### **REGISTER WRITE and INCREMENT WRITE**

w [i2c address] [register] [data\_1] [data\_2] ... [data\_k]  
> [data\_k+1] ... [data\_m]  
> [data\_m+1] ... [data\_n]

Where 'i2c address', 'register' and 'data\_x' are in hexadecimal format.

NOTE: n is less than or equal to 32 for a GUI communicating through the TAS1020B.

---

#### **REGISTER READ**

r [i2c address] [register] [read amount]

Where 'i2c address', 'register' and 'read amount' are in hexadecimal format and read amount is less than or equal to 32 (0x20).

---

#### **COMMENT**

# [any text]

Lines beginning with the # character are ignored.

---

**INTERFACE (EVM GUI ONLY)**

**i** [interface]

Where 'interface' is:

i2cstd	Standard mode I2C Bus
i2cfast	Fast mode I2C bus
spi8	SPI bus with 8-bit register addressing
spi16	SPI bus with 16-bit register addressing
gpio	Use the USB-MODEVM GPIO capability

---

**BREAKPOINT (EVM GUI ONLY)**

**b** ["string"]

Where "string" can be any string of characters to be displayed at a pop-up message by the GUI.

---

**DELAY**

**d** [milliseconds]

Where 'milliseconds' is the delay time in decimal format.

---

**WAIT FOR FLAG**

**f** [i2c address] [register] [D7][D6][D5][D4][D3][D2][D1][D0]

Where 'i2c address' and 'register' are in hexadecimal format and 'D7' through 'D0' are in binary format with values of 0, 1 or X for don't care.

---